Particle Trajectory Classification and Prediction Using Machine Learning

Angelos Angelopoulos (CRTC), Polykarpos Thomadakis (CRTC), Gagik Gavalian (JLab), Nikos Chrisochoides (CRTC)
Department of Computer Science, Old Dominion University, Norfolk, VA, 23529

Abstract

Nuclear physics is a challenging scientific domain where experiments are often expensive due to the high cost of the machinery involved. Experimental setups record terabytes of data each day and process them to identify interacting particles from information provided by a series of detectors. One of the most important parts of data processing is identifying trajectories of charged particles in wire chambers. This process is computationally expensive and comprises about 94% of computational time. Charged particles are identified by combinatorially considering all possible combinations of segments. In this work, we used machine learning to identify possible valid combinations of track segments to reduce the number of combinatorics to be considered and reduced the computational time by a factor of ~6.

We developed three different models to address this problem: an extremely randomized trees model, a multi-layer perceptron as well as a convolutional neural network (CNN). The models achieved an overall classification accuracy of 96.5%. To further reduce the search space for classification, we developed a supervised recurrent neural network (RNN) using gated recurrent unit (GRU) layers capable of predicting particle trajectories based on previous trajectory information. Because the model is trained on only valid trajectories (i.e. broken lines), it will help eliminate many invalid trajectories that do not align with its predictions. These machine learning models will be employed in the experimental pipeline for the CLAS12 detector in order to filter incoming data, save 6-8x more time and energy compared to current methods used, and help increase experimental accuracy.

Introduction

The CLAS12 detector at Jefferson Lab is used to study the structure of matter by scattering an electron beam off a proton target. Particles produced as a result of the interaction are detected by the signal a particle leaves in wire drift chambers. In order to reconstruct the trajectories, all possible combinations of segments need to be considered before accepting the one that most closely fits the shape of a "broken line." Because this combinatoric computation is very computationally intensive, a faster and reliable method was needed to identify valid particle trajectories.

Because speed is crucial in the experimental pipeline, we developed machine learning models that can classify incoming particle trajectory data with very high throughput through multi-core inferencing as well as GPU-accelerated inferencing. Our filtering solution consists of two components: prediction of particle trajectories based on previous trajectory information and classification of trajectories as being valid or not.

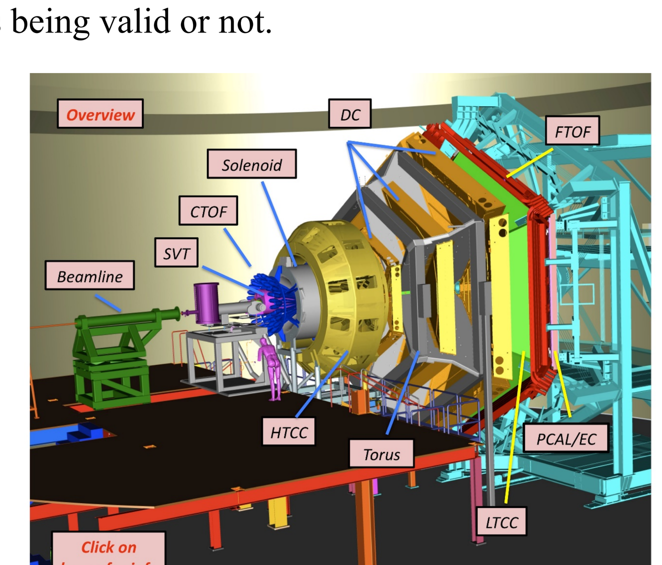

Figure 1 The CLAS12 detector in Jefferson Lab, Experimental Hall B.

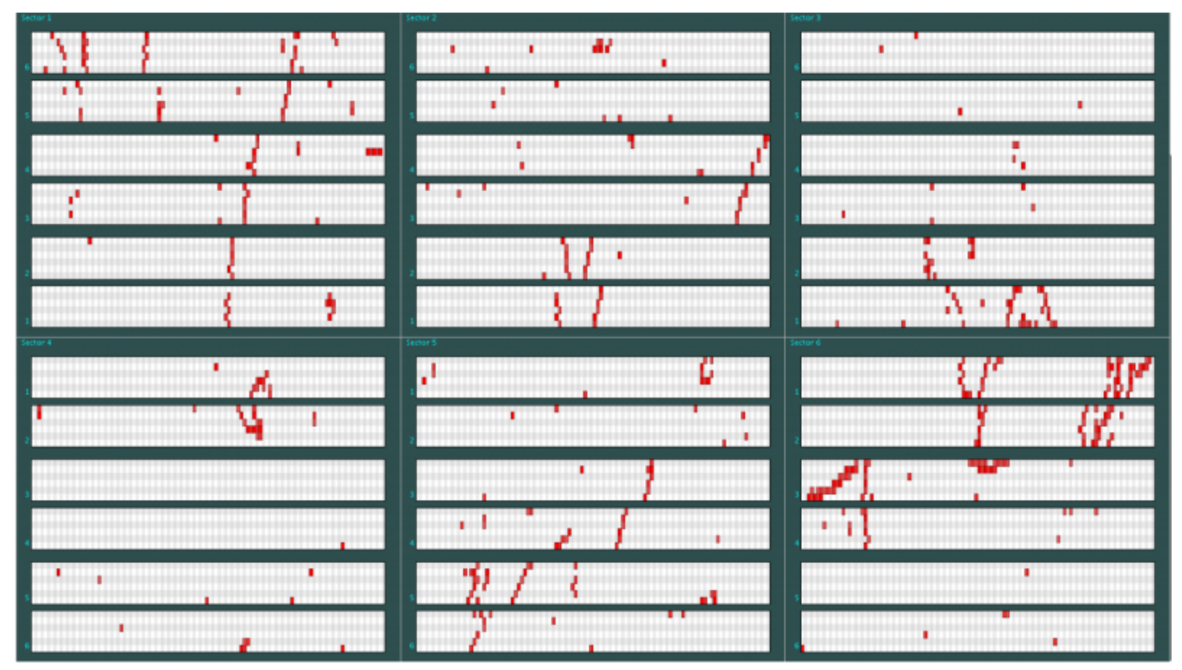

Figure 2 CLAS12 detector drift chamber showing the sensors activated (red) by traveling charged particles. The drift chambers of the CLAS12 detector consist of 6 layers, each of which contain 6 wires (total 36 wires). Each wire contains 112 sensors, for a total of 4032 (36*112) sensors. The activations that form an approximate broken line from beginning to end are classified as valid.

Method

The two problems we had to solve are particle trajectory classification and particle trajectory prediction. Particle trajectory classification involves determining if a valid particle track is present in experimental data while particle trajectory prediction involves predicting particle trajectories based on previous partial trajectory data.

Particle Trajectory Classification. In order to solve this problem we developed three separate supervised machine learning models: an extremely randomized trees model (ERT), a multi-layer perceptron (MLP), and a convolutional neural network (CNN). The ERT and MLP models were developed using the *scikit-learn* library and run on the CPU while being able to utilize multiple processing cores. The CNN model was developed using *Keras/TensorFlow* and runs on the GPU. Training and evaluation of these models utilized tens of thousands of labeled particle trajectory samples from CLAS12 detector experiments. Samples included multiple particle tracks, of which one was valid. All these models were developed to have high performance, as they need to be capable of real-time inference and high-throughput. The MLP model classifies particle trajectories as valid or not in 4 µs, the ERT model in 5 µs, and the more complex CNN model in 1.2 ms.

In order to determine our models' accuracy, we devised and utilized several metrics in addition to using the standard accuracy metric in scikit-learn and Keras. These new accuracy metrics are:

1. **A1**: The ratio of samples where the valid particle track was correctly detected.
2. **Ac**: The percentage of A1 for which there were invalid tracks misidentified as valid ones (false positives).
3. **Ah**: The percentage of A1 for which the valid particle track had the highest probability of being valid out of all tracks in a sample.
4. **Af**: The ratio of samples where the valid track was not detected (false negatives). This metric was very critical for us to minimize, as we don't want to miss valid particle tracks.

Particle Trajectory Prediction. Because predicting a particle trajectory based on incomplete previous trajectory information involves completing a sequence, we utilized a supervised recurrent neural network (RNN) using gated recurrent unit (GRU) layers. The RNN is trained on the same dataset as the models for trajectory classification, except that only data for valid particle tracks is utilized. This allows the RNN to predict valid particle tracks based on partial previous sensor activation patterns. Because the RNN is trained on only valid particle tracks, it will give incorrect predictions for invalid particle tracks. By measuring the spatial distance of the actual particle track and the one predicted by the RNN, we can infer whether the actual particle track is valid or not. A large distance likely means that the particle track is invalid, whereas a small distance likely means that the particle track is valid as it aligns with the model's predictions. This allows us to eliminate many samples where there are no valid tracks present.

Since particle trajectory prediction happens before classification, we reduce the number of samples containing invalid tracks that are to be classified, reducing the likelihood of incorrect classification and thus increasing overall accuracy and eliminating more invalid tracks.

Results

Our machine learning models have high performance and high overall accuracy. For particle trajectory classification, we achieved an overall accuracy of 96.5%. For particle trajectory prediction we achieved an overall loss of ~1.18 (loss in this case is defined as the mean distance in sensors of predictions from the actual data). Figure 3 shows two valid particle tracks and the predictions made by the RNN. Figure 4 shows an invalid particle track and the prediction of the RNN. Notice how the prediction of the RNN for the invalid particle track is wrong and spatially distant from the actual track. A larger distance likely means that the track is invalid. Table 1 shows metrics for the trajectory classification models, while Table 2 shows metrics for the RNN.

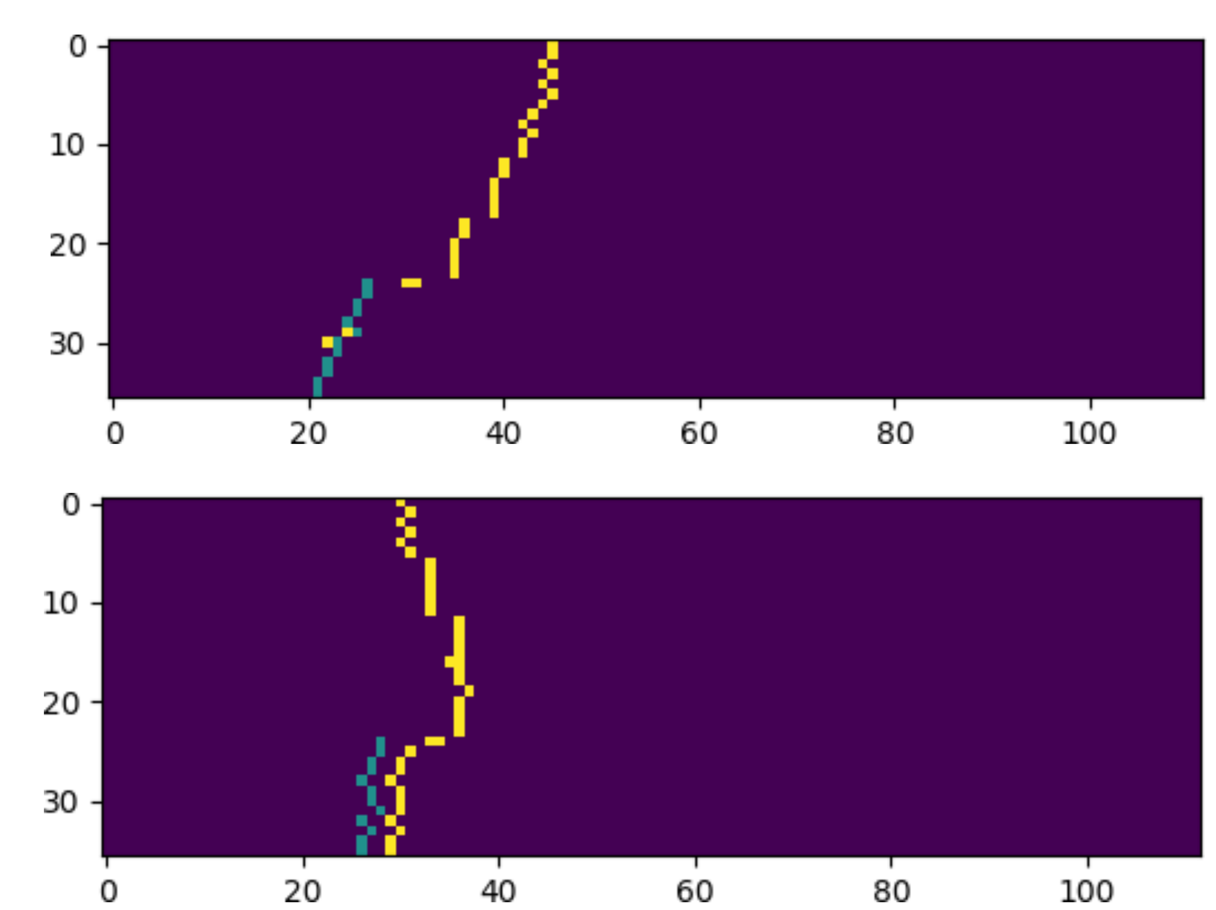

Figure 3 Shows two separate *valid* actual particle tracks (yellow) and the predictions of the RNN for part of them (blue). If the RNN correctly predicted the track, then the yellow and blue overlap (as seen in the first image). The small spatial distance between the predicted portion of the tracks and the actual portion of the tracks means that the actual tracks are likely valid.

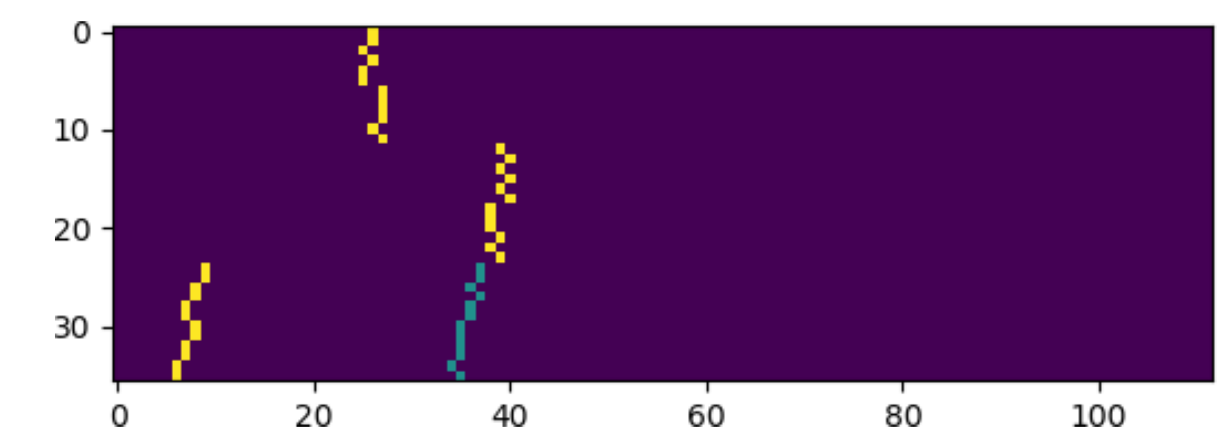

Figure 4 Shows one *invalid* actual particle track (yellow) and the prediction of the RNN for part of it (blue). The large spatial distance between the predicted portion of the track and the actual portion of the track means that the actual track is likely invalid.

Model Type	*A1* Metric	*Ac* Metric	*Ah* Metric	*Af* Metric	Training Accuracy	Time to Train	Time to Predict / sample
MLP	96.5%	20.2%	92.1%	3.4%	94.7%	252 *sec*	4 *µs*
ERT	93.3%	19.9%	91.9%	6.6%	99.9%	1.7 *sec*	5 *µs*
CNN	96.4%	30.1%	89.4%	3.5%	93.4%	457 *sec*	1.2 *ms*

Table 1 Metrics for our three models for particle trajectory classification. The metrics are described in the "Method" section. MLP and ERT executed on a multi-core CPU while the CNN executed on one Tesla V100-SXM2-16GB.

Model Type	Loss (MAE)	Time to Train	Time to Predict / sample
RNN/GRU	~1.18	374 *sec*	688 *µs*

Table 2 Metrics for our RNN for particle trajectory prediction. The unit of the loss is one sensor. So, a loss of 1.18 means a mean distance of 1.18 sensors between the actual and predicted particle tracks. The RNN executed on one Tesla V100-SXM2-16GB.

Conclusion

The machine learning models we developed have high accuracy and high throughput. By employing them in the experimental pipeline for the CLAS12 detector at Jefferson Lab to filter incoming data and discard invalid particle tracks, we can save 6-8x more time and energy compared to current methods used, and ultimately increase the accuracy of experiments.

Acknowledgements

This work is in part funded by JLab, SURA grant No. CNF-19-04, NSF grant no. CCF-1439079, and Richard T. Cheng endowment. This poster describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the poster do not necessarily represent the views of the United States government.